Priya Aggarwal and Shreya Rajeswaran
DS3000 Section 4
29 November 2023
Final Report

Event Price Predictions Using the Ticketmaster API

Abstract

Our paper investigates the pricing dynamics of recent concerts in the Boston area through data generated with the Ticketmaster API. We first had to clean the data taken from the API as some events were missing critical information/had conflicting pieces of information. The analysis involved creating a linear regression model utilizing the features genre, season, ticket limit, and venue size of a specific event to predict the event's average ticket price. With the model, we were able to explain about 61 percent of the variance in average price using the features we collected. To create an accurate model using linear regression, we had to employ techniques such as scale normalization of numeric features, creating indicator variables for categorical features, and taking the logarithmic function of the average price.

Introduction

Ticketmaster is an online platform where tickets for events may be distributed and bought. They connect with various performance venues and artists to be the primary avenue to sell tickets for events (primarily concerts) on their website and also allow resale capabilities for individual people who have bought tickets to sell them to others. While some of the responsibility of setting ticket prices lies with the event coordinators and the public, Ticketmaster utilizes their own algorithms to set prices for events.

The pricing dynamics of events have changed drastically post-pandemic with concerts being a staple social event in modern times ("Why Are Concert Tickets"). Concerts have been becoming more expensive over the years because of the increase in popularity, with tickets regularly going on sale for hundreds of dollars initially. As ticket prices increase, fans are looking for less expensive events to attend while still enjoying the concert experience. There is much uncertainty of what causes certain events to be more expensive than others, as ticket prices widely range between different types of concerts.

Ticket prices are not usually released until after the tickets go on sale. With more analysis into what factors contribute to ticket prices, fans can be informed in their purchasing decisions and better identify their events within their price range. They may also identify whether a concert is appropriately priced based on its features or whether Ticketmaster is attempting to upsell the ticket. This information may also help to stabilize resale prices since fans will have a better understanding of what constitutes a fair price for an event ticket in the current market.

There are many questions that could be answered looking at Ticketmaster API data, such as:

- What variables contribute/relate most to the price of tickets? Potential variables to look at include venue size, promoter, and ticket limit.
- What genres tend to have more or less expensive events? For example, pop events might have the highest average ticket price.
- What variables lead to a venue being on average more expensive compared to another venue?
- Can we determine whether a concert is properly priced compared to other concerts? This involves predicting the price using other variables and seeing how far it deviates from our model's expectations.

Using the cleaned data, we will attempt to answer some of the questions we proposed above. For the analysis, we will have to employ some machine learning models.

Primarily, we could use a multiple regression model using some of the numeric and categorical variables to predict the price of a concert using its features. With the correlation coefficients produced by the model, we could analyze what variables have the largest correlation coefficient and contribute most to the price as well as analyze which the variables have positive and negative correlation with price. This model would most likely use genre, ticket limit, venue size, and season as its features.

Another model we considered was a logit model (classification) for determining if a venue is an expensive venue or not. This would involve deciding what the ticket price is that divides an expensive venue versus a cheap venue. We could potentially find data about average ticket prices in Boston to make this measurement less subjective. With this classification, we can

use the variables given by Ticketmaster to create a prediction model for whether a venue will be expensive or not.

We also will use k-fold or Leave-One-Out cross validation when creating these models to assess its accuracy and avoid overfitting the data. This would involve splitting the data between training and testing during cross-validation and fitting the model to the whole data set in the end.

Figure 1. *Ticketmaster API Schema*

Data Description

In this project, data was gathered from the Ticketmaster API, which includes information about a wide variety of features for different events (see Figure 1).

This API is fairly reliable as it is tied with Ticketmaster itself, which is a well-known vendor for various event tickets. It is also well-documented which adds to its reliability and accessibility.

To narrow the scope of our research, we decided to focus on music-related events that occurred in Boston, and updated the url accordingly with those restrictions. We grabbed a random subset of 200 events because the API restricted us to that many per call.

After obtaining this data, we loaded it into a JSON dictionary and indexed it to obtain various features of interest, such as venue name, event name, average price, season, genre, ticket limit, venue size, and whether or not there was an age restriction. We decided to try to obtain as many features as possible so that we had a lot of options for the machine learning methods and could try different combinations to see what might work the most effectively even if we didn't use all this information in the models.

Some of the factors that we had to consider were:

- Some events did not have dictionary keys for price or other features at all. To solve this, we used a try/except block to add information about that event if it was present, or add a null value if that information was missing.
- Some events had an age restriction present in the title of the event such as “18+” or “21+”, despite the actual dictionary value for age restriction actually being False. So, we manually went through and checked event titles for the presence of these age restrictions and updated the value in the dataframe accordingly.
- The information about ticket limit was present in a string (ex. “There is an 8 ticket limit for this event”) so we needed to write a function to extract an integer from a string.
- We wanted to obtain season information in our dataset, so we wrote a function to convert a string date into a datetime object and then figure out what season that date belonged to.
- The API did not include information about venue size, so we created a dictionary by finding the venue size of key Boston venues online and incorporating that information into our dataframe.

Our final step to clean the data was dealing with the NaN values. We observed that the only missing data was average price and ticket limit; since we felt that both of these features were important for our analysis, we decided to drop all events with missing data for either of them. There were also only a few values that were null so we still had a significant amount of data leftover (186 out of 200 events originally).

Figure 2. *Histogram of average event prices vs log average prices*

To visualize the data, we first wanted to observe the distribution of average price because it is the variable we are predicting and we wanted to determine if it would work with the models we are hoping to use.

However, after observing the graph, we saw that the prices were very right-skewed. This suggested to us that transforming the price column through taking its log could help us get a more Normal distribution which could be better for analysis (Figure 2).

Next, we wanted to explore the relationship between genre and avg_price. We decided to group by genre and plot a bar chart of each genre's average avg_price across all events (Figure 3).

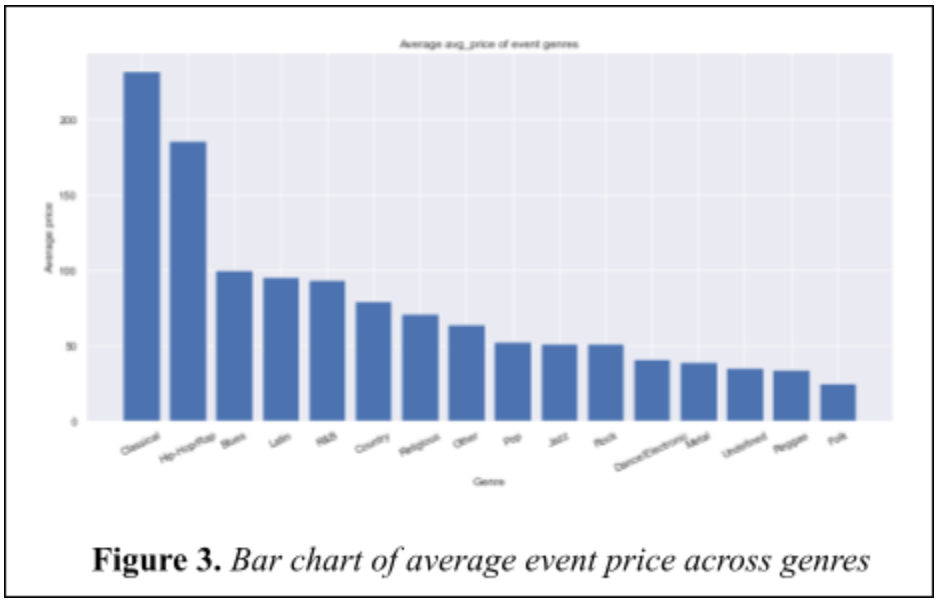

Figure 3. *Bar chart of average event price across genres*

After observing this visualization we can see that classical and hip hop or rap events are usually a lot more expensive than reggae or folk. We were surprised that pop events were so low because we feel like they are usually expensive; however, maybe famous artists play in venues just outside of Boston that are larger, but are not considered "Boston" locations. Upon looking at our dataset more closely, we also noticed that there were only 1-2 classical and hip hop/rap concerts and they happened to be expensive. This indicates that there are perhaps too few instances of events in these categories to assume that these are expensive categories.

Method

In this project, we decided to implement the linear regression machine learning model to this dataset as a means of predicting price based on the other features of an event. This method is appropriate because regression is a tool that is used to predict numerical variables like price. It

calculates the line of best fit through the relationship between the X variables and the y variable (price) by finding the line that will minimize the mean squared errors (MSE) between the predictor and response variables.

To implement this model, we made sure to check assumptions of residual independence, constant variance, and normality to ensure that a linear fit would be appropriate for the data and for the x variables we were using in the regression.

Before we began, we made dummy variables for some of the categorical features in order to be able to use them in the regression. We scaled the numeric variables so that units do not play a role in the weight of the variable in the ML model. Additionally, we cross-validated using single fold (70-30 split) and LOO-CV so that our model would be training and testing data as it developed. LOO-CV was effective since our dataset was not too large and it is a more robust way of cross validating. We used a multiple regression model while also being conscious of not overfitting the model to the data.

While creating dummy variables, we tried to stay aware that if there were too many features and not enough data observations, this could lead to large p small n issues and would require the implementation of dimensionality reduction or PCA to our data. We also needed to be careful when choosing the features we add to the regression model, since many of our features like venue_name and venue_size could predict each other, and this would lead to multicollinearity within our regression.

A limitation we observe with our data is that we were not able to obtain the performer name. There is no key in the original dictionary API that includes the performer's name, and the event titles are inconsistent with the way they mention performers, if at all. We know that the person who is performing likely plays a big role in how much an event costs, but we will not be able to use it in our analysis.

Results

To begin our implementation of our linear regression machine learning model, we began by selecting variables we thought would be the most apt to include. We decided to select two categorical variables (genre and season) as well as two numeric features (ticket limit and venue

size) as we believed that they would most significantly impact the average price of an event out of the other variables we had obtained in our overall dataframe.

We began by obtaining dummy variables for genre and season to be able to implement it in the machine learning model. However, when examining the genre more closely, we noticed that there were some genres with only one or two events in the dataset which we felt was not significant enough to use as a predictor variable for this regression. So, we decided to drop all the genre dummy variables with 3 or fewer events in the dataset. We examined the model that was created with all the genres vs with the reduced number of genres and observed that there was no significant change in R2, so since the addition of those genres did not have a big impact on the model, this further confirmed that we should drop them.

Next, we scaled the numeric variables ticket limit and venue size by dividing them by their standard deviations. This was important because we wanted to ensure that neither of them held more weight in the machine learning model because of a difference in magnitude of units. We used np.hstack to put all the predictor variables together in an X matrix and obtained the average price as our response variable in a numpy array. We ended up using the log-transformed average price, for reasons that will be discussed further in the discussion.

Pictured below are our residual plots.

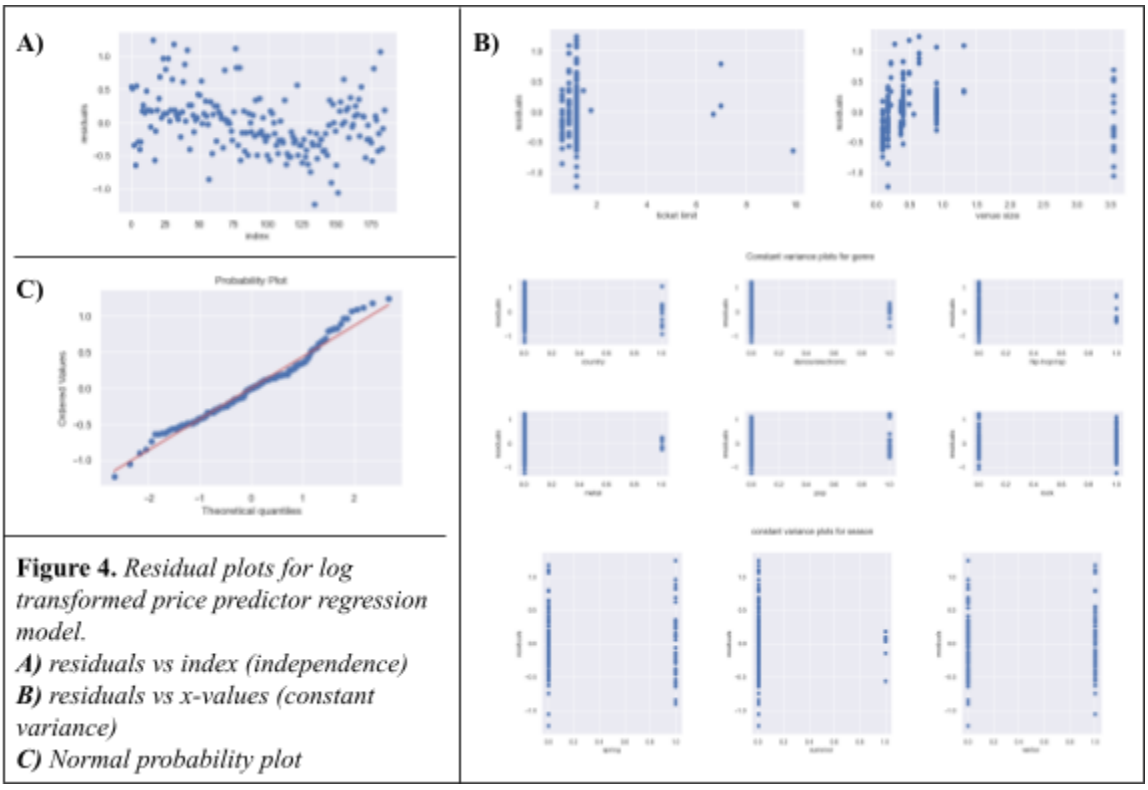

Figure 4. *Residual plots for log transformed price predictor regression model.*
A) *residuals vs index (independence)*
B) *residuals vs x-values (constant variance)*
C) *Normal probability plot*

Additionally, the coefficients we obtained for this plot were:

```
In [34]: model.coef_

Out[34]: array([-0.0233339 ,  0.49549938,  0.07197984, -0.15274507,  0.07947168,
                -0.17943127, -0.36956856, -0.09562411,  0.252962  ,  0.34416158,
                 0.138917  ])

In [35]: model.intercept_

Out[35]: 3.4512790687240105
```

The order of the coefficients is the following: ticket_limit, venue_size, genre_country, genre_dance/electronic, genre_hiphop/rap, genre_metal, genre_pop, genre_rock, season_spring, season_summer, season_winter.

Discussion

Initially, we did not implement a log transformed average price.

When we interpreted this initial model that predicted average price, we cross-validated the data using both single fold (70-30) and LOO-CV and obtained an R2 of 0.485 and 0.490, respectively. This told us that the model was moderately strong at predicting the price of an event based off of the predictor variables we inputted. Additionally, since this is cross validated, it suggests that our model is not overfitting the data, so we progressed to fitting the model to the overall data. This led to an overall R2 value of 0.611 which suggests that 61.1% of the variation in average price of an event can be explained by the features we chose to predict it.

However, when we looked at the residual plots (Figure 5), this is where we realized the model was not meeting the assumptions of a regression model.

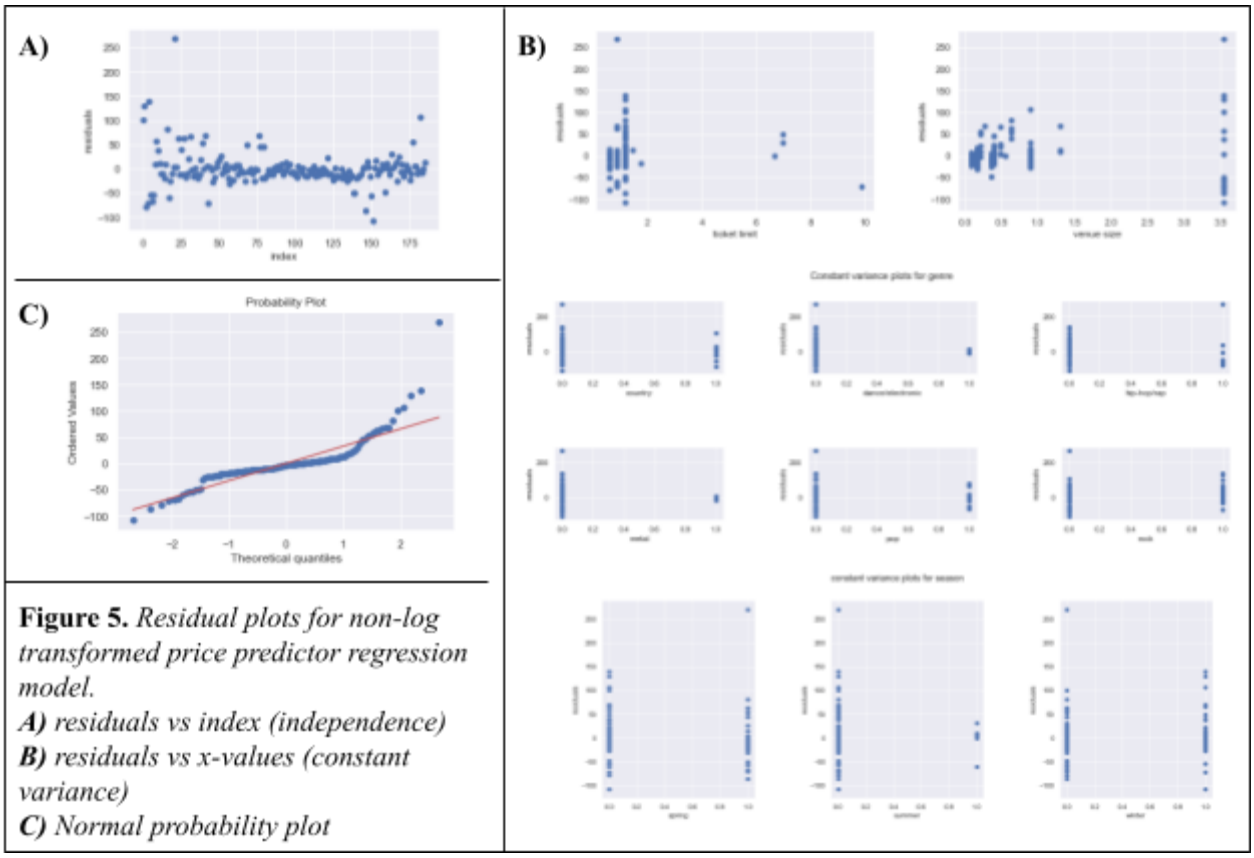

Figure 5. *Residual plots for non-log transformed price predictor regression model.*
A) *residuals vs index (independence)*
B) *residuals vs x-values (constant variance)*
C) *Normal probability plot*

The first residual plot we looked at was simply graphing residuals by their index value to examine how scattered they are. For the most part, they looked randomly scattered around 0. There were some notable outliers with one residuals reaching a value of over 250. This outlier also made the chart much more scrunched together around 0, making it difficult to analyze the rest of the values. We also wanted to create residual plots comparing our different features to their associated residuals. However, since most of our variables were either categorical or numeric with few distinct values, it is difficult to examine the randomness of the variables compared to their features. For venue size, it seems like the outliers were larger for larger venue sizes compared to smaller, disrupting our assumption of homoscedasticity. Finally, we looked at the probability plot to see if the residuals were normally distributed. They seemed to be for the most part; however, it wasn't completely fleshed to our normal distribution expectations.

To remedy some of the failed assumptions, we decided to try to use the transformed price using log rather than the normal price. The R2 for this model using LOO-CV (0.549) was similar

to the original model, and so we fit the model to the whole dataset and obtained an R2 of 0.614, meaning that the model is successful at explaining 61.4% of the variance in log average price based off of the x features. We proceeded to create residual plots (see Figure 4) of the model when fitting to the whole dataset. Primarily, when looking at the residual plot of the residuals ordered by index, we see a much more scattered distribution with less notable outliers. Additionally, the venue size residual plot and other features look more randomly scattered as well. Finally, residuals are much closer to normal distribution using the log transformation. This makes sense because the y variable is no longer as right skewed with the transformation and is also closer to a normal distribution. Thus, we decided that it was more meaningful to create our linear regression model to predict the log(avg_price) values instead of the original.

To interpret the results of our model for use, we can take a look at the coefficients and intercept of the model in context. Note that the model is predicting the log transformed version of average price rather than average price itself. Primarily, we can analyze the numeric features' coefficients. The ticket limit has a coefficient of -0.023 meaning that a larger ticket limit will lead to a cheaper concert. This makes sense in context since Ticketmaster may distribute less tickets for higher demanded concerts. The other numeric feature, venue size, has a coefficient of 0.495. This is the largest coefficient within our model and means that a larger venue means a more expensive concert. This also makes sense in context, since larger venues tend to attract bigger performers and have more costs that require the higher ticket prices.

Next, we move into the categorical features. It is important to understand the intercept of the model when analyzing the categorical features. The intercept of the model is 3.451. This means that the expected log transformed ticket price of a concert would be about 3.45 dollars if the ticket limit and venue size was 0 (which does not have much meaning in context). This also requires that the genre is one of the dropped/smaller genres within our model and that the season is fall, since these were the categories not explicitly included in the model.

Looking at the explicitly defined categorical variables that were represented as dummy variables within the model, we can find the coefficients for the following genres: 0.072 for country, -0.153 for dance/electronic, -0.079 for hip hop/rap, -0.179 for metal, -0.370 for pop, and -0.096 for rock. Most of these coefficients are negative, meaning that a change in genre from the other category to one of these would lead to a decrease in price. This goes against our initial expectations that pop and rock concerts would be the most expensive. However, we think this

makes sense according to our dataset since we found that some of the smaller and more obscure genres showcased one or two very expensive events like classical.

Finally, we can look at the coefficients of the seasons to see how the time of year impacts the price of a concert. The coefficients for seasons are the following: 0.253 for spring, 0.344 for summer, and 0.139 for winter. This means that a change from Fall (the current season) to any other season would lead to an increase in the price of a ticket, with summer having the most expensive concerts. This makes sense in context since summer is a popular time to attend concerts and this contributes to the higher price.

If someone were to use this model to find cheaper concerts, they would want a concert with a higher ticket limit, happening in the Fall, genre pop, and located in a small venue.

While the R2 value of our model is promising, it is important to mention various limitations we faced when creating a model with our dataset. One limitation was that Ticketmaster could not provide an artist name for the events despite us filtering by music-related events. This is because Ticketmaster is a platform used to sell tickets for all different types of events and did not have an "artist" category to store this information. Before using this dataset, we were hoping to use the artist and artist popularity (sourced through the Spotify API) to help predict the price of an event ticket. The inability to use this data was a major limitation within our model.

Additionally, our dataset had a lack of numeric features to use. The two numeric features used were ticket limit and venue size. While these variables were numeric, they were discrete and not continuous and thus had few distinct values. A lack of interesting numeric data ruled out some models such as the Linear Perceptron. It also made it more difficult to visualize our linear regression model on our dataset, since even the graphs with numeric features had only a couple sections (noted previously when discussing the residual plots). This also makes the model more difficult to interpret and explain to others without a more complete understanding of our dataset, since the graphs are not easy to understand. Furthermore, it was difficult to identify if there were polynomial relationships between the data because of the discrete nature of the few numeric variables we implemented.

It is also important to note our difficulty when extracting data through the Ticketmaster API. The API only delivered 200 events at a time, creating a smaller dataset than previously

desired. Additionally, it was difficult to impact the randomness of the API's event selection, so we ended up with a large amount of events with the genre rock compared to other genres.

Work Cited

Kaplan, Ilana. "Why Are Concert Tickets so Expensive in 2023?" *Peoplemag*, PEOPLE, 29 July 2023, people.com/why-are-concert-tickets-so-expensive-in-2023-7567208.